Structure landmark-based self-supervised tracking

Shu Liu and Jianjie Zhao

Research School of engineering, Australian National University,

Canberra ACT 0200 Australia

E-mail: u5491675@anu.edu.au

u5764321@anu.edu.au

Abstract

Deep neural network is a mature technology to discover the higher-level features from the higher dimension data. However, there are quite few researches about tracking objects with little annotation. In this paper, a detailed solution of tracking the critical elements in a video with self-supervision has been proposed. We employ the optical flow algorithm to crop the target object in images by detecting the concentrated motion field. We further proposed a self-supervised landmark detecting method to discover the critical spatial features of the target. Two hourglass networks are combined to form an encoder-decoder architecture for capturing the predominating features. Ideally, such captured feature can structurally represent the tracking target. In the experiment, we test our method on two different datasets. Although the results are not satisfactory on the dataset with complex scene since there are lots of uncorrelated information in the background; while for the simpler dataset, the method has a relatively better performance. In general, we design and implement an innovative self-supervised tracking method, and verify its performance on different datasets.

Keywords

Self-supervised, Deep convolutional neural networks, object tracking, structural representation

1. Introduction

The feature learning of intrinsic properties, such as shape and size is one of the most difficult challenges for computer vision. Although in images, the appearance of objects highly depends on the intrinsic structures. Some other unpredictable factors like viewpoints can also affect the states of the objects[1]. Therefore, it is extremely difficult to detect the structure of objects from images. One of the possible solutions is to manually annotate the structure such as landmarks and skeletons. But the fact is that manual annotations are really pricey and seldomly available[2]. Hence, it is valuable to automatically model the intrinsic viewpoint-independent structure of object.

The deep neural networks (DNN), as a matured technology to extract high level feature from the raw data, play a more and more important role in the image processing. In [3], the authors employed a large deep convolutional neural network to solve the classification problem of the 1.2 million images with 1000 different classes. Also, a fully convolutional networks can be used for the sematic segmentation[4]. Moreover, some other common aspects in computer vision such as object detection[5], 3D reconstruction[6] can be implemented by the deep neural networks. However, there is not much information about

efficiently using the networks to model the intrinsic structures of object.

In this project, we aim to apply a new approach to learn the object structure and position few supervision. And as a commonly used technology to illustrate the shape of the object, landmarks are visible and can also provide the spatial information of the main parts of objects. In [1], the author provided a strategy for the unsupervised learning of object landmarks. This method is based on the image deformations by factorizing the different viewpoints of the object. The main drawback is the landmarks found by the networks are not on the critical location such as face, body. While in our work, the landmarks are encouraged to appear on the target objects.

Since we start with an optical flow to extract the significant motion from the complex raw image. The optical flow can help to track the motion of a certain object [7]. In this project, the tracking target, ant, is cut out from the frames with its motion field. The optical flow method provides an effective approach to filter out the uncorrelated parts from the raw image.

2.related work

Motion tracking. Object tracking is one of the biggest subjects in the computer vision. In general, feature matching is the most popular algorithm for the motion tracking [7]. The author in [8] provides a correlation filter to implement feature matching which improves the processing speed. The optical flow field is used to show the object motion between image pairs, which is defined as the displacement of pixels in image [7]. Lucas-Kanade method and Horn-Schunck method provides the theoretical support of generating optical flow [9]. Scene optical flow is proposed based on the optical flow [10]. Different from optical flow which focuses on the two-dimensional motion field, scene flow is the three-dimension motion field [10]. More recently, FlowNet successfully apply the deep convolutional neural network (DCNN) on optical flow [11]. It employs an hourglass-style architecture network which is widely used in the following flow estimation project.

Features of objects. There are many different possible methods to model the intrinsic structure of objects [12-14]. And it has been proven that the modelling of features has wide applications such as object detection, facial landmark detection and pose estimation. Most of the work in the literatures is derivative of object detection system and, it is usually in a supervised deep learning system. In [15], the author built unsupervised system for a spatial transformer network which can find the geometric transformations of the objects.

Decomposition of the important features from an object is considered as alternative method to model the structure of objects. From the work in [16], a vision object can be decomposed into many different variations such as camera viewpoint, pose and motion state. The parameters of variations can be embedded into latent representations.

Unsupervised learning. In unsupervised learning, there is no ‘teacher’ to guide the learning process. In other word, the data given in unsupervised learning are unlabelled. Autoencoders and denoising autoencoder are common traditional methods for unsupervised learning [18]. They can learn the useful features from the input data and reconstruct the original data by some restrictions. The generative adversarial network (GAN) [19] aims to find the generative models to produce samples of images. The author in [19] employs GAN to find the latent representations to extract important information from images. In recent years, many studies suggest that unsupervised learning can be used for the auxiliary tasks and, the supervision can be achieved without any manual operation. These kinds of methods employ parts of existing information as input. And the trained networks are required to reconstruct the remaining data. It often be considered as ‘self-supervision’. In [20], Noroozi and Favaro use a novel unsupervised

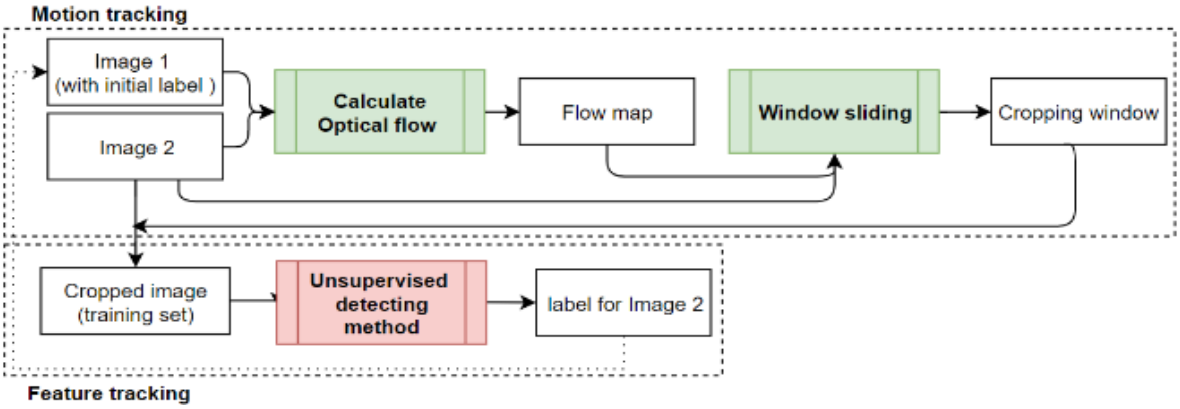

Figure 1 architecture of the tracking system

learning approach to solve jigsaw puzzles of natural images.

3.Method

3.1 Architecture of tracking system

We are inspired by the visual real-time human tracking system [22, 23] and propose a general architecture for solving tracking problem with a video input, which is shown in Figure1.

Our designed model takes a few of frames in a video as input. The initial position $p_1 = (\mathrm{x}_1, \mathrm{y}_1)$ of the tracking target is annotated on the first frame. By defining a size of cropping window r, the four coordinates of the vertexes of the cropped square image $\boldsymbol{S_1}$ in the first frame can be expressed as $(x_1 - r)$, $(x_1 + r)$, $(y_1 - r)$, $(y_1 + r)$. In this case, the position of the cropped image $\boldsymbol{S_1}'$ in the second frame can be determined with the same coordinates.

An optical flow map is then calculated based on the two cropped images $\boldsymbol{S_1}$ and $\boldsymbol{S_1}'$ from the first frames, while the most concentrated motion is captured by a sliding window $\boldsymbol{w}$. The centre coordinate of the $\boldsymbol{w}$ is then taken for updating the coordinates of cropped image $\boldsymbol{S_2}$. Under such circumstance, we can iteratively obtain a new window $\boldsymbol{S_n}$ based on $n-1^{th}$ and n^{th} frames. And the cropping window covers the potential region of tracking target.

The motion tracking with pure optical flow cannot follow the target for a long while. The cropped images need to be manually sorted in order to ensure that all the output images keep containing the tracking target

In this case, we only need to find the first cropped image in which the target is missing and excludes images later. Even though there are large amount of cropped images extracted in one video. The manual sorting work is not time-consuming.

As for the training set, the tracking target is continuously appearing in all the cropped images, so that the tracking target can be identified without extra annotation. In the other words, since the tracking target is the only object which predominates in all cropped image, an unsupervised method would figure out the precise position of the predominating target in the patches, while the position in n^{th} frame is denoted as $p_n = (\mathrm{x}_\mathrm{n}, \mathrm{y}_\mathrm{n})$.

In this case, our architecture only requires the annotated position of the target in the first frame, which results in a iterate training. Ideally, no further annotation is required.

In general, our architecture applies optical flow method for generating a series of candidate cropped images from video frames. Such cropped images are manually sorted for feeding an unsupervised model to capture the critical feature of the tracking target.

Besides, what should be noticed is that our architecture is restricted with the following preconditions.

1. The target should have a relatively obvious motion compared with the background.
2. The target should occupy a large proportion area in the cropped images.

At the same time, the architecture has the following benefits compared with the current tracking methods:

1. The optical flow produces a cropping window which shrinks the searching region of the target.
2. The supervised method minimizes the annotation requirement.
3. The motion tracking section reject the static background, which enhances the performance of the system in a complex scene.

3.2 Local window searching

For updating the position of the cropping window, a window $\boldsymbol{w}$ is sliding across the cropped image for capturing the most concentrated motion. Since the optical flow is calculated based on two cropped images, the flow map should has the same length r with the cropped image. In this case, we denote the optical map as K and define the length of sliding window to be a. The operation of sliding window can be mathematically expressed by:

$$(\mathrm{x},\mathrm{y}) = \mathrm{argmax}(\sum_{i=\frac{a}{2}}^{r-\frac{a}{2}}\sum_{j=\frac{a}{2}}^{r-\frac{a}{2}} K(i,j))$$

Meanwhile, since this sliding window operation is conducted though the whole image, it means the coordinate of the centre of cropping window (x,y) will be affected by a large motion that is far from the position in the previous image.

Under such circumstance, for employing the speed constraint, we assume a maximal pixel-wise displacement d_x, d_y of tracking target in the image. And the window is sliding in the range of $[-d_x,\ d_x]$ and $[-d_y,\ d_y]$ instead of $[\frac{r}{2}, X-\frac{r}{2}]$ and $[\frac{r}{2}, Y-\frac{r}{2}]$ for x-axis and y-axis respectively.

3.3 Separation of training process

Data imbalance is a critical problem for the training of our architecture. As the position of the tracking target and the feature of background are predominated in the recent frames, they would misconduct the model.

In this case, an offline training method is proposed by employing more manual annotations. We exclude the feature tracking section in Figure1 and take pure optical flow for iteratively predicting the position of target, which can help to generate cropped images consisting of the tracking target.

Meanwhile, it is noted that target would be lost by the window in a few of frame if only the first frame is annotated. Here, we empirically define index of starting frame i and the maximal number of frames n, while the tracking target should be in the region of the cropping window between i^{th} and $(i+n)^{th}$ frames.

Algorithm1 Training set generating

Define maximal frame $\boldsymbol{n}$
Define frame index $\boldsymbol{i}$ = 0
Define training set Λ as an empty list
Input Image sequence I_n from a video, where n is the index
Input Initial position of target in the i^{th} frame (x_i, y_i)
Input maximal displacement d_x, d_y
While i < n:
    Calculate the optical flow map K based on I_i and I_{i+1}.
    Capture the most concentrated motion on K with a window sliding (d_x, d_y) far from the position(x_i, y_i), the centre position and side length of the window denotes as $(x_w, y_w) = (d_x + x_i, d_y + y_i)$ and α.
    Cropping the squared patch I_i' in the region of $(x_w \pm \frac{a}{2}, y_w \pm \frac{a}{2})$ from the I_i.
    Append I_i' into Λ as a training sample.
    Update (x_i, y_i) with the values of (x_w, y_w)

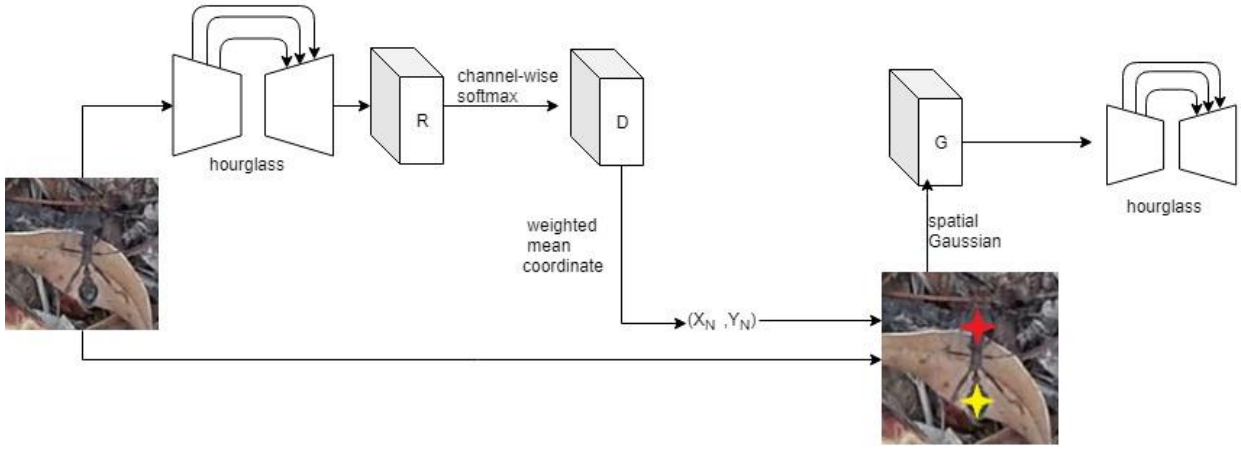

Figure 2 flow chart for the self-supervised network

The pseudocode demonstrates the process of generating n training samples with n continuous frames, which consists of one kind of scene.

In this case, if m pairs of $(i, (x_i, y_i))$ are prepared, by repeating Algorithm1, it will output mn training samples with m different background scenes.

Besides, since the generating of dataset is independent from the training period. The methods of data augmentation such rotation and flip are valid for enlarging the dataset.

3.4 Landmark detector

The locating of the landmarks is just the problem of detecting key features in the image. In this project, we intend to let evert landmarks be controlled by the detector. The detector is able to generate a score map for the potential landmarks and the finally confirmed landmark is the one with the maximum score. In [21], the author successfully use an hourglass network to estimate the human pose. Hence, as can be seen in Figure2, the hourglass-style network is employed to transform the input image to a $(k + 1)$ channel with detection score map R. The first k channels output k landmarks and the $k + 1$ channel is the background of the image.

$$\mathrm{R} = \mathrm{hourglass}(\mathrm{input\ image})$$

The hourglass-style architecture can also constrain the detector to pay more attention on the critical location in the image. However, the different detectors are possible to find the same landmark. To avoid the duplication, we aim to transform the unrestricted scores from map R into the probabilities. Therefore, the soft-max method is used to achieve the final detection score map.

$$\mathrm{D_k(a,b)} = \frac{\exp(\mathrm{R_k}(a,b))}{\sum_{k=1}^{k+1}\exp(R_k(a,b))}$$

Where $\mathrm{D_k}$ is the $\mathrm{k_{th}}$ channel of D. D is corresponding to module D in Figure2. (a, b) is the pixel. $\mathrm{D_k}(a, b)$ is the result score for $\mathrm{k_{th}}$ channel of D at the pixel (a, b).

Then the coordinate of the landmarks can be defined as the weighted mean of the map $\mathrm{D_k}$.

$$(\mathrm{x_k}, \mathrm{y_k}) = \frac{1}{\sum_{b=1}^{B}\sum_{a=1}^{A} D_k(a,b)} \sum_{b=1}^{B}\sum_{a=1}^{A}(a,b)D_k(a,b)$$

The term $\sum_{b=1}^{B}\sum_{a=1}^{A} D_k(a,b)$ is just the spatial normalization factor. The set of landmarks can be rewritten as

$$\mathrm{landmark(I)} = [(\mathrm{x_1}, \mathrm{y_1}), (\mathrm{x_2}, \mathrm{y_2}), \ldots, (\mathrm{x_k}, \mathrm{y_k})]$$

After getting the landmarks, the image with landmarks at the critical location can be formed.

The left half of figure 2 illustrate the landmark detection process. The landmarks with different detectors are represented by diverse colors.

Landmark Constraint

Although the landmark coordinates can be discovered by the algorithm mentioned above, the unsupervised learning leads to a random position of the landmarks. That is to say, the landmark can be located at any position in images rather than on the target object. Hence, the following constraint are required.

Concentration constraint. The mass of D_k is supposed to be concentrated in the local region. We first define the spatial normalization factor s_k.

$$s_k = \sum_{b=1}^{B} \sum_{a=1}^{A} D_k(a,b)$$

Then a density of the bivariate distribution is formed by $\frac{D_k}{s_k}$. And the variance σ_a^2, σ_b^2 along the x and y axis of $\frac{D_k}{s_k}$ are computed. The variance is required to be as small as possible. Therefore, the concentration constraint loss is defined as:

$$L_c = (\sigma_a^2 + \sigma_b^2)^2$$

Distance constraint. In general, in order to reconstruct the original input image, the k landmarks discovered by the CNN are supposed to be at different positions. However, these k landmarks are not guaranteed to cover the target object. In another word, if there is no restriction, the k landmarks are highly possible to be located around the same point. Hence, a distance loss is introduced to disperse the landmarks.

$$L_D = \sum_{k \neq k'}^{1,\dots,K} exp(-\left|\left|(x_k', y_k') - (x_k, y_k)\right|\right|_2^2)$$

3.5 Decoder

The right half of figure 2 is the reconstruction of the input image. The detection score map R is recovered by the landmark coordinates. Each landmark is enlarged by an isotropic Gaussian distribution.

$$\mathrm{G_k}(a,b) = \mathcal{N}((a,b); (x_k, y_k), (\sigma_a^2, \sigma_b^2))$$

While these Gaussian distribution are feeding another hour-glass network for reconstructing the image. A mean squared error E_{mse} is calculated between the reconstructed image and the input image. In general, the overall loss is

$$E = E_{mse} + \lambda_c L_c + \lambda_d L_d$$

where λ_c and λ_d are the weight of different regularization terms.

3.6 Ant-in-wild and Ant-on-ball Dataset

Our unsupervised solution aims to track a target in a complex scene. The restriction of our method limits its performance on common dataset. In this case, our algorithm is tested on the unlabelled data of ant motion which are provided by ANU Zeil lab.

Ant-in-wild (shown in Figure8) data is a bottom view video with a length of 57 seconds and 25 frames per second. In the video, there are two ants moving slowly, while the background includes stones, weeds, sticks and leaves. In such scenario, distinguishing and tracking motion of ant is a challenge.

Figure 3 sample image in ant-in-wild

Ant-on-ball (shown in Figure9) is a dataset consists of motion of ant which is fixed on a ball by a needle. The ant can arbitrarily move but cannot leave the region of the white ball. There are a few patterns on the ball, which can be considered as noise. It is noted the background has much less spatial features that would have negative influence on prediction compared with Ant-in-wild. There are 5403 images in this dataset, while the image size is 971x736.

	Dimension	Dataset Size	Background	Target Moving	Annotation
Ant-in-wild	2160x3840	1425	Noised	Unrestricted	None
Ant-on-ball	971x736	5403	Non-noised	Restricted	None

Table 1 summary details of dataset

Figure 4 sample image in ant-on-ball

In summary, Ant-in-wild demonstrates a tracking problem in real scenario that the background contains complex spatial features which could confuse the detector. Ant-on-ball is a relatively simple dataset, which has less background noise and a target with restricted motion.

As for Ant-in-wild, we test our tracking system that takes a 100x100 cropping window for tracking the moving ant. While for Ant-on-ball, since the moving region is ant is limit, we only test the unsupervised method.

3.7 Image proprocessing

Due to the hardware limitation, we prune the number of channels of image for reducing the computational cost. In this case, we convert the RGB channels into one channel, while the input dimension is changed from $(x_{in}, y_{in}, 3)$ to $(x_{in}, y_{in}, 1)$.

Also, image rotation and flipping is necessary for enlarging the training dataset and enhancing ability of detecting target feature of model, as there are limited different degree of ant body orientation included in our dataset.

4. Results and discussion

4.1 Landmark discovery on complex scene

The test of our unsupervised method on Ant-in-wild is based on an offline training. In such case, we manually capture two ants with the same size from the video and assume the maximal frame $\boldsymbol{n}$ to be 250, which contributes to a dataset which consists of 500 training samples.

These sample images are split into training set and test set with a ratio of 9:1. By setting a 128x128 cropping window, we conduct the method described in Algorithm1 for generate cropped images. In most case, the optical flow catch the ant, as the ant conducts the most concentrated motion in the scene.

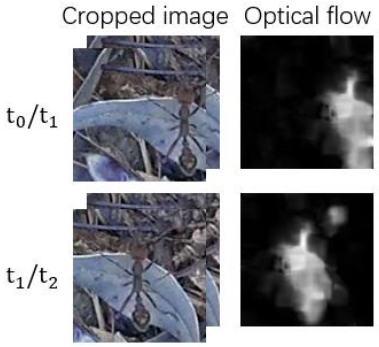

Figure 5 the raw images as well as optical flow map.
The motion field in optical flow map fits the position of the ant.

Meanwhile, the optical flow map would become unclear if the ant is static, and this method is unrobust to the noise in the scene, such as swing grass or foliage, while we will lose the path of ant with these background influences.

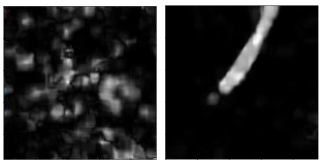

Figure 6 the unexpected feature map

In this case, we mannually clean the image set, and select 382 cropped images for training.

As for the unsupervised learning, we empirically set the weight of regularization λ_c and λ_d to be 0.01 and 0.001 respectively, while the number of critical points is 5.

The training cost 4 hours for process 200 epochs on a desktop with GTX1060 graphic card.

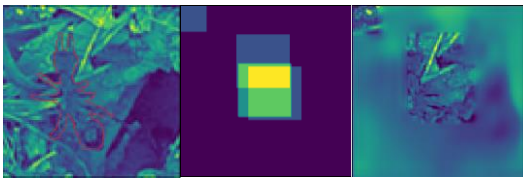

Figure 7 results on Ant-in-wild dataset. Raw image(left). ant is marked by red. Windows(middle). Reconstructed image(right)

Although most of the window are covering the body of ant, we can find that if the ant is apart from the centre of the cropped image, window seems inaccurate. And we furtherly conduct statistical analysis to the position of these predicted windows.

Table 2 Average the centre position of the points

Index	*1*	*2*	*3*	*4*
x-axis	*63.71*	*65.62*	*0*	*66.54*
y-axis	*63.78*	*69.68*	*0*	*74.78*

Table 3 Standard derivation the centre position of the points

Index	*1*	*2*	*3*	*4*
x-axis	*0.40*	*3.69*	*0*	*11.85*
y-axis	*0.41*	*7.21*	*0*	*10.63*

From the Table2 and Table3, we can find although we take 5 windows for reconstruct the image, all windows only change its position with a standard derivation of less than 10 unit. Considering that the image have a size of 128x128, these windows seems static across the testing set.

Also, the positon of these windows seems irrelative to any spatial feature of ant. It can be reflected from Figure7, in which the none of the centre of the windows occurs on the body of ant, we also cannot figure out the appearance of ant from the reconstructed image. In this case, we can suppose that the model does not learn the effective landmark representations on the body of ant.

4.2 landmark discovery on clear scene

For the pre-processing of Ant-on-ball dataset, we remove the black edges and only keep the spatial information on the ball. The channel of image is also reduced during training process, and the dimension is 128x128x1. As for the parameters, we reserve the same setting with the pervious test.

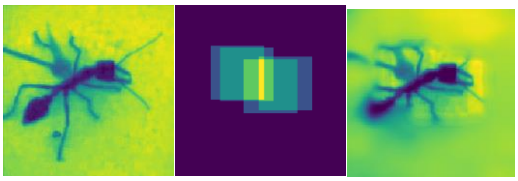

Figure 9 results on Ant-on-ball dataset. Raw image(left). Windows(middle). Reconstructed image(right)

As for the performance, we can find that the reconstructed ant image (the right image in Figure 9) is distinguishable. As for the windows, they all locate on the body of ant, which shows that the landmark representation can capture our target if the background consists of less noise.

Table 4 Average the centre position of the points

Index	1	2	3	4
x-axis	64.00	67.74	65.49	66.55
y-axis	71.18	68.55	66.62	62.09

Table 5 Standard derivation the centre position of the points

Index	1	2	3	4
x-axis	15.26	6.90	15.62	3.88
y-axis	16.35	29.54	16.68	16.86

The statistical analysis in Table 4 and Table5 shows that all the windows have a relatively large moving region, which means their position are changing with different ant position.

In Figure11, more predicted result are demonstrated that the windows somehow capture the components on ant body.

5. Conclusion

In this paper, we propose an adaptable tracking system and apply a self-supervised method for capturing the critical landmark of the target. We

further test our method on two image dataset of ant tracking, which shows the feasibility of our method with a few annotations if the scene is simple.

As for the failure on Ant-in-wild, one of the key reasons is the lack of images. Although there are 500 images generated for training, such images are all based on just two scenes. Under such circumstance, the failure is surprising, as the ant would not be the predominating object in such dataset. Generating more data with different scenes would be necessary in this case.

Our work can also be considered as an extended application of the unsupervised structural representation. Meanwhile, the regularization can be more target-oriented for tracking an appointed target. With some prior knowledge, the colour or the spatial pattern can be converted to a regularization term, which can improve the performance on tracking a specific target.

6. Acknowledgement

We would like to express our thanks to Dr. Trevor Murray as well as other staffs in ANU Zeil lab for offering the images data of ant.

7. Reference

[1] J. Thewlis, H. Bilen, and A. Vedaldi, "Unsupervised learning of object landmarks by factorized spatial embeddings," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5916-5925.

[2] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee, "Unsupervised discovery of object landmarks as structural representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2694-2703.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, pp. 142-158, 2015.

[6] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366-2374.

[7] P. H. Torr and A. Zisserman, "Feature based methods for structure and motion estimation," in *International workshop on vision algorithms*, 1999, pp. 278-294.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, pp. 583-596, 2014.

[9] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *International journal of computer vision*, vol. 61, pp. 211-231, 2005.

[10] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 722-729.

[11] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, *et al.*, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758-2766.

[12] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-

their training and application," *Computer vision and image understanding*, vol. 61, pp. 38-59, 1995.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *international Conference on computer vision & Pattern Recognition (CVPR'05)*, 2005, pp. 886--893.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, pp. 1627-1645, 2009.

[15] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017-2025.

[16] S. Reed, K. Sohn, Y. Zhang, and H. Lee, "Learning to disentangle factors of variation with manifold interaction," in *International Conference on Machine Learning*, 2014, pp. 1431-1439.

[17] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, pp. 1499-1503, 2016.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, pp. 504-507, 2006.

[19] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[20] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision*, 2016, pp. 69-84.

[21] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*, 2016, pp. 483-499.

[22] Q.cai and J.K.Agrawal "Human tracking motion in structured environment using distributed camera system" IEEE transactions on pattern analysis on machine intelligence volume 21 number 12 november 1999

[23] M.-J. Seow D. Valaparla V. K. Asari "Neural network based skin color model for face detection" Proc. Appl. Image Pattern Recognit. Workshop pp. 141-145 2003

8. Appendix

Figure 10 results on Ant-in-wild dataset. Raw image(left). Windows(middle). Reconstructed image(right)

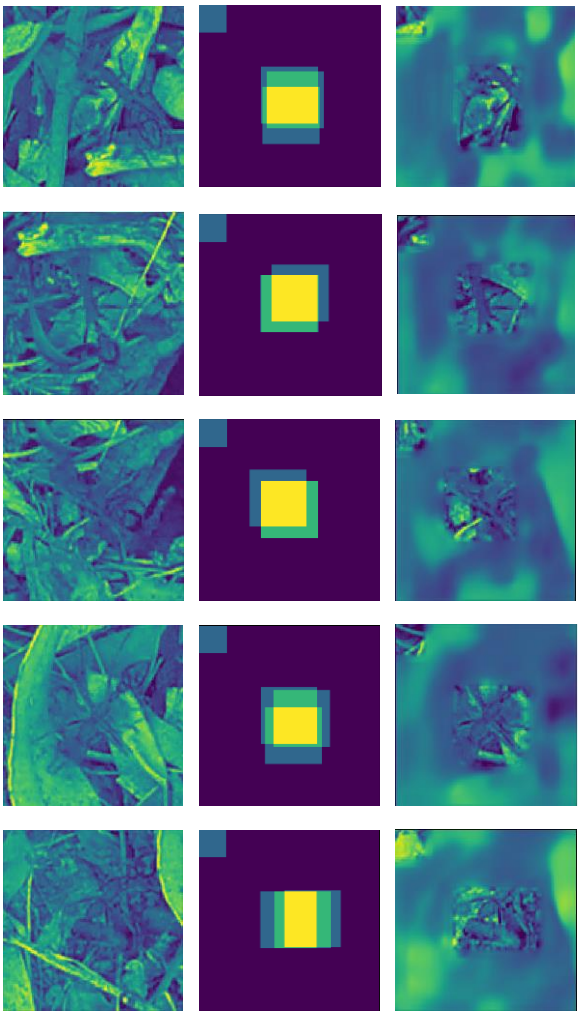

Figure 11 results on Ant-on-ball dataset. Raw image(left). Windows(middle). Reconstructed image(right)